Post Login Issues

Category: Troubleshooting

Idle SSH Connections are Broken After 10-15 Minutes

If you find that idle SSH connections tend to be dropped after 10 to 15 minutes and your are using a router or other device that implements Network Address Translation (NAT), such as a DSL router, you may want to enable the **`ServerAliveInterval`** parameter in your **`~/.ssh/config`**.

The issue is that some devices that implement NAT will drop information from their state tables if a TCP connection has been idle for just a few minutes. In such cases, it may be necessary to ensure that the connection is never truly idle for an extended period.

The **`ServerAliveInterval`** parameter (supported in OpenSSH 3.8 and later) causes the SSH client to periodically send an encrypted query to determine if the server is still responsive. The packets will make the connection appear to be active and can prevent the NAT state table entry for the connection from timing out.

For example, adding the following to your **`~/.ssh/config`** will cause an encrypted query to be sent every five minutes:

```
ServerAliveInterval 5m
```

TIP: You may need to experiment with the length of the interval to determine an appropriate value to prevent your connection from being broken. If you use the NAS config template, **`ServerAliveInterval`** is set to 10 minutes.

SSH Connection is Periodically Broken Even When Connection is Active

During an SSH session, the client and server will re-negotiate the keys used to encrypt the session every so often. Very old versions of OpenSSH did not implement this feature, which can cause a problem when connected to the SFEs. Your connection may appear to hang or may abort with an error message such as the following:

```
Dispatch protocol type 20
```

If you encounter this issue, you will need to upgrade to a newer version of OpenSSH. Note: Other implementations that were derived from the OpenSSH distribution, such as SUN Secure Shell, are likewise known to exhibit this problem.

Slow Performance Transferring Data

While OpenSSH performs reasonably for local data transfers, the performance will tend to be reduced due to the latency of long-haul network connections.

Depending on the severity of the impact you may have several options to improve this situation.

Upgrade to OpenSSH 4.7 or Later

If you are using a version of OpenSSH that is older than 4.7, you may see an improvement in file transfer performance by upgrading to OpenSSH 4.7 or later. This is due to the use of a larger channel buffer introduced in that version.

Use an HPN-enabled Version of OpenSSH

The High-Performance Networking (HPN) patch set maintained by the Pittsburgh Supercomputing Center (PSC) allows the channel buffer used by OpenSSH to grow as needed. This may be useful in cases where OpenSSH 4.7 does not yield satisfactory performance.

In cases where you are seeing very poor performance (under 1.2 MB/s) and the data you are transferring will compress well, you may wish to enable compression.

You can do this by adding **`-C`** to your **`scp`** or **`sftp`** command-line.

Read Tips for File Transfers for more recommendations.

Article ID: 238
Last updated: 25 Jul, 2012
Troubleshooting -> Post Login Issues
http://www.nas.nasa.gov/hecc/support/kb/entry/238/?ajax=1